

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-083710

(43)Date of publication of application : 25.03.1994

(51)Int.CI.

G06F 12/10

(21)Application number : 05-006915

(71)Applicant : INTERNATL BUSINESS MACH CORP <IBM>

(22)Date of filing : 19.01.1993

(72)Inventor : CLARK CARL E  
FREY JEFFREY A  
PLAMBECK KENNETH E  
SCALZI CASPER A  
SINHA BHASKAR

(30)Priority

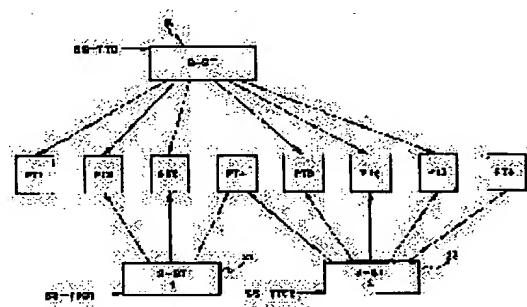
Priority number : 92 847521 Priority date : 06.03.1992 Priority country : US

## (54) PROGRAM AND DATA PROCESSING METHOD

### (57)Abstract:

PURPOSE: To provide an efficient inter-space branch process to separated programs in different address spaces in a group of related address spaces of a subsystem.

CONSTITUTION: The group has a fundamental space and at least one partial space, and each partial space can include a program and data, which cannot be accessed from other partial spaces, to separate the program and data in a subsystem. Any CPU in a computer system can designate programs found in address spaces. A designating process checks authority for access to the group of a user of the group and executes an intra-group program call instruction which designates first and second general registers.



## LEGAL STATUS

[Date of request for examination] 21.05.1993

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-83710

(43)公開日 平成6年(1994)3月25日

(51)Int.Cl.<sup>5</sup>

G 0 6 F 12/10

識別記号 廈内整理番号

E 7608-5B

F I

技術表示箇所

審査請求 有 請求項の数16(全 18 頁)

(21)出願番号 特願平5-6915

(22)出願日 平成5年(1993)1月19日

(31)優先権主張番号 8 4 7 5 2 1

(32)優先日 1992年3月6日

(33)優先権主張国 米国(US)

(71)出願人 390009531

インターナショナル・ビジネス・マシンズ・コーポレイション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州  
アーモンク(番地なし)

(72)発明者 カール・エドワード・クラーク  
アメリカ合衆国12603、ニューヨーク州ボ  
ーキブシー、パート・ドライブ 46

(74)代理人 弁理士 順宮 幸一(外4名)

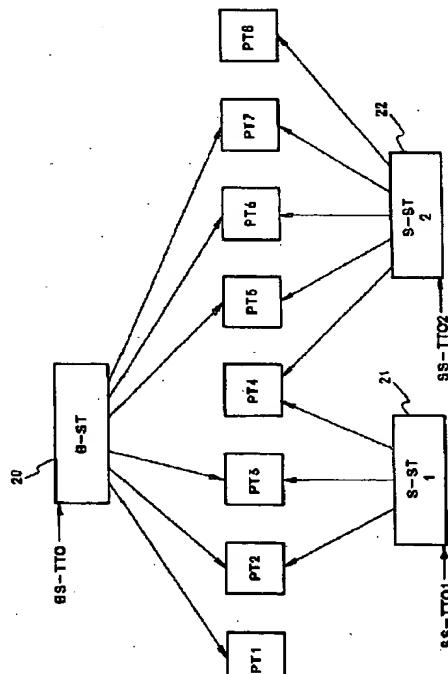
最終頁に続く

(54)【発明の名称】 プログラム及びデータの処理方法

(57)【要約】

【目的】サブシステムの関連アドレス空間のグループ内の異なるアドレス空間中の分離されたプログラムへの効率的な空間間の分岐プロセスの提供。

【構成】前記グループは基本空間及び少なくとも1つの部分空間を有し、各部分空間は、サブシステム内のプログラム及びデータの分離を可能にする、他の部分空間からはアクセスできないプログラム及びデータを含むことができる。コンピュータ・システム内のCPUはどれもアドレス空間で見つかったプログラムを指名できる。前記指名するプロセスはグループのユーザの前記グループをアクセスする権限も検査し、第1及び第2の汎用レジスターを指定するグループ内プログラム呼出し命令を実行する。



1

## 【特許請求の範囲】

【請求項1】サブシステム内のプログラム及びデータを効率的に処理する方法であって、  
プログラム及びデータを含む関連アドレス空間のグループとして基本アドレス空間(基本空間)及び部分アドレス空間(部分空間)を供給するステップと、  
プログラム及びデータを他のアドレス空間でアドレス指定できないようにすることにより1つのアドレス空間のプログラム及びデータが全ての他のアドレス空間からの間違った記憶動作から保護されるようにアドレス指定関係を構築するが、部分空間ページの一部又は全部に対し基本空間が一部又は全部の部分空間と共通のアドレス指定能力を持ち得るよう所要の予め計画され限定された共通アドレス指定能力を可能にするステップと、  
基本空間内のページのアドレス指定能力を与える基本空間変換テーブルを取得しあつ各個の部分空間内のページのアドレス指定能力を与えるそれぞれの部分空間変換テーブルを取得するステップと、

部分空間グループの基本空間と全ての部分空間によりアドレス指定されたページを特定し、又は共有のオペレーティングシステムにより実行中の他のサブシステム又はプログラムによる使用から限定された共通のアドレス指定関係を有する共有のプログラム及び(又は)データを含むページ・フレームを有する基本空間変換テーブル及びそれぞれの部分空間変換テーブルの両者によりそれぞれのページフレームを取得することにより部分空間グループの基本空間及び部分空間の間でアドレス指定能力を共有することを可能にするステップと、

更に部分空間に限定されたアドレス指定能力を有する部分空間内で実行するプログラムを供給するが、基本空間内で実行中のプログラムにより部分空間に対するアドレス指定能力を供給するステップとを含むサブシステム内のプログラム及びデータを効率的に処理する方法。

【請求項2】前記構築するステップは前記グループ内の基本空間及び各部分空間のためにそれぞれのセグメント・テーブルを供給し、前記基本空間のセグメント・テーブルは部分空間セグメント・テーブルと共通するページ・テーブルをアドレス指定するエントリを取得し、そして各ページ・テーブルは共有のプログラム及び(又は)データを含む共通のページをアドレス指定するエントリを取得するステップを含む請求項1のサブシステム内のプログラム及びデータを効率的に処理する方法。

【請求項3】前記構築するステップは前記グループ内の基本空間及び各部分空間のためにそれぞれのページ・テーブルを供給し、前記基本空間のページ・テーブルは部分空間ページ・テーブルと共通するページ・フレームをアドレス指定するエントリを取得し、そして各共有ページ・フレームは共有のプログラム及び(又は)データを包含するステップを含む請求項1のサブシステム内のプログラム及びデータを効率的に処理する方法。

2

【請求項4】前記構築するステップは変換テーブルがアドレス空間のグループ内のアドレス指定能力を供給する時期を表示するために各変換テーブルに関連した変換テーブル記述子内にグループ・フィールドを供給するステップを含む請求項1のサブシステム内のプログラム及びデータを効率的に処理する方法。

【請求項5】前記構築するステップはプログラム呼出し命令に対するユーザの従来の権限検査に少なくとも部分的に依存するグループ内プログラム呼出し命令を実行し、前記プログラム呼出し命令は第1及び第2の汎用レジスタを指定し、前記第2のレジスタは分岐される宛先アドレスを含み前記第1のレジスタは前記プログラム呼出し命令の直ぐ後に続く命令のアドレスを受取るステップと、

前記第2のレジスタに関連したアクセス・レジスタにある部分空間グループ内の宛先アドレス空間を指定するステップとを含む請求項1のプログラム・サブシステム内のプログラム及びデータを効率的に処理する方法。

【請求項6】サブシステム内のプログラム及びデータを効率的に処理する方法であって、  
基本アドレス空間(基本空間)及び関連アドレス空間(部分空間)を供給することによりプログラム・サブシステムの関連アドレス空間のグループを構築し、前記基本空間及び部分空間は前記サブシステムのプログラム及び(又は)データを包含し、そして前記プログラムが前記グループ内の基本空間及び部分空間に共有されるステップと、

プログラム呼出し命令のユーザの従来の検査に少なくとも部分的に依存するグループ内プログラム呼出し命令を実行し、前記プログラム呼出し命令は第1及び第2の汎用レジスタを指定し、前記第2のレジスタは分岐される宛先アドレスを包含し、そして前記第1のレジスタは前記プログラム呼出し命令の直ぐ後に続く命令のアドレスを受取るステップと、

前記第2のレジスタに関連したアクセス・レジスタにある部分空間内の宛先アドレス空間を指定するステップとを含むサブシステム内のプログラム及びデータを効率的に処理する方法。

【請求項7】グループ内のアドレス空間で最後に実行された部分空間及び基本空間の識別子により見つかった、コンピュータ・システムのCPUの制御手段をロードする動作を含む、プログラムを実行するタスクを指名し、前記CPUで前記グループ内の前記タスクにより最後に実行されたプログラムが基本空間にあるか部分空間にあるかを示す標識を取得するステップを含む請求項6の異なるアドレス空間内の分離されたプログラムに効率的に分岐する方法。

【請求項8】前記識別子及び標識を含む指名可能装置制御テーブル(DUCT)のアドレス(DUCTO)をCPUにロードすることにより、コンピュータ・システム内の指名中のCPU

50

内の制御手段として制御レジスタを提供するステップを含む請求項6の異なるアドレス空間内の分離されたプログラムに効率的に分岐する方法。

【請求項9】コンピュータ・システムのメイン・メモリ内の指名可能装置制御テーブルを見つけるステップを含む請求項6の異なるアドレス空間内の分離されたプログラムに効率的に分岐する方法。

【請求項10】前記指名するステップの一部としてユーザが前記グループにアクセスすることを可能にするユーザ権限を検査するステップを含む請求項6の異なるアドレス空間内の分離されたプログラムに効率的に分岐する方法。

【請求項11】前記指名されたプログラムの識別子を前記基本空間の識別子と比較するステップと、前記プログラム呼出し命令が前記グループ内のアドレス空間にあるプログラム間でプログラム制御を移すためにのみ動作するという制約を強制するために、前記比較するステップが前記比較された識別子間の一致を認める場合にのみグループ内プログラム呼出しの実行を継続できるようにするステップとを含む請求項6の異なるアドレス空間内の分離されたプログラムに効率的に分岐する方法。

【請求項12】異なるアドレス空間内の分離されたプログラムに効率的に分岐する方法であって、基本空間及び部分空間を有するサブシステムの関連アドレス空間のグループを構築し、前記基本空間及び部分空間は前記サブシステムのプログラム及びデータを包含するステップと、

前記グループ内のアドレス空間で見つかったプログラムを実行するために、ユーザが前記グループをアクセスできるユーザ権限を検査しあつ前記CPU内の制御レジスタに前記グループの最後に実行された部分空間及び基本空間の識別子を含む指名可能装置制御テーブル(DUCT)のアドレス(DUCTO)をロードすることによりコンピュータ・システム内のCPUにタスクを指名し、そして前記グループ内の最後に実行されたプログラムが基本空間にあるか部分空間にあるかを表示する標識を取得するステップと、

第1の汎用レジスタと、分岐される宛先レジスタを含む第2の汎用レジスタと、前記第2のレジスタ内の宛先アドレスを含む宛先アドレス空間を指定するために前記第2のレジスタに関連したアクセス・レジスタを指定することにより、前記指名するステップによるユーザ権限の検査に依存する前記指名されたプログラム内のグループ内プログラム呼出し命令を実行するステップと、前記指名されたプログラムの識別子を前記DUCT内の前記基本空間の識別子と比較するステップと、前記プログラム呼出し命令が前記グループ内でのみ動作する制約を強制するために前記呼出し中のプログラムが前記グループ内にあることを検査する前記比較ステップ

が一致を見つける場合にのみプログラム呼出しの実行を継続することができるステップと、

前記第2のレジスタに関連したアクセス・レジスタの内容によって決められたグループ内の宛先アドレス空間にある記憶ブロックの実アドレスを得るために、宛先アドレスを前記第2のレジスタ内の仮想アドレスからアドレス変換するステップとを含む異なるアドレス空間内の分離されたプログラムに効率的に分岐する方法。

【請求項13】前記グループ内の他のプログラムにより共通に用いられるプログラムを前記ページ・フレームが含み、前記プログラムが、指定されたレジスタに前記命令復帰アドレスを保管しているグループ内に他のプログラム及びデータを含みうる他の記憶キーを割当てられた記憶ブロックをアクセスすることとなっていないとき、記憶ブロックに共通記憶キーを割当てるステップを含む請求項12のグループ内の異なるアドレス空間内の分離されたプログラムに効率的に分岐する方法。

【請求項14】現在のインスタンスの実行のために当該空間が開始されたとき前記最後に実行された部分空間を決めるASN第2テーブル・エントリ(ASTE)のシーケンス番号の標識をCPU内に供給し、次にそのシーケンス番号がASTEのシーケンス番号値により現在の品質について検査された後に、前記CPU内の新たな第1のアドレス空間又は第2のアドレス空間を設定するために前記CPU内に記憶されたASTE0(ASTE原点)を用いるステップを含む請求項12のグループ内の異なるアドレス空間内の分離されたプログラム間で効率的に分岐する方法。

【請求項15】前記アドレス空間のASTEを無効にする(前記アドレス空間のASTE内のシーケンス番号を増分する)ステップと、

無効になったアドレス空間を指名できないように第1及び第2のアドレス空間を再設定することを各CPUに要求するステップとを含む請求項12のアドレス空間を使用する前にアドレス空間を無効にする方法。

【請求項16】前記グループ外のアドレス空間内のプログラムに制御を移すために部分空間グループの基本空間又は部分空間内のプログラムにより空間間プログラム呼出し命令を使用し、次のプログラム復帰命令が前記グループ内の前記呼出すアドレス空間に制御を戻すステップを含む請求項12の部分空間グループ内の異なるアドレス空間内の分離されたプログラム間で効率的に分岐する方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、仮想部分空間内のデータ(プログラムを含む)を、同じ部分空間グループ内の他の仮想部分空間にあるプログラムから分離することにより、コンピュータ・システムの動作の信頼性を向上することに関する。仮想アドレス変換テーブルは異なる部分空間ならびにそれらの基本空間のアドレス空間をそれぞ

5

れ与える。共通の変換テーブル・エントリを有する変換テーブルを備えることにより、基本アドレス空間は部分空間に対するアドレス指定能力 (addressability) を有する。従って、基本空間内で実行するプログラムはその部分空間に対する所定のアドレス指定能力を有するが、部分空間内のプログラムは必ずしも他の部分空間に対する、又は基本空間のアドレスの全てに対するアドレス指定能力を有しない。前記アドレス指定能力の分離は、部分空間のデータ及びプログラムを、他の部分空間による偶然のアクセス (取出し及び記憶) から保護する。部分空間のアドレス指定能力の範囲、従ってアドレス指定能力の分離は、該部分空間を排他的に分割する、基本空間内の共通の変換テーブルによって確立される。

## 【 0 0 0 2 】

【 従来の技術】一般に従来の技術では、プログラム及びデータの族 (ファミリ) は、それらの関係及び実行がからみ合うので、異なるプログラムの間の高速切換えが必要であり、かつ族内でデータベースが共用される。このようなプログラム及びデータの族はたいてい (オペレーティングシステムの下に動作する) ソフトウェア・サブシステムによってサポートされる。このサブシステムはたいてい族内の多数の異なるプログラム及びデータベースを同時にアクセスしようとする多数のトランザクションを処理する。例えば、多重分岐バンキング・コンプレックス (bankingcomplex) におけるバンキング・テラー (人間及び機械の両者) の同時トランザクションは、個々のサービス要求により呼出されたトランザクション・プログラムのセットによって同時にアクセスされている、預金プログラム及び引出プログラム (同じデータベース、即ち顧客の預金口座を共有する) 、クレジット照合プログラム及びそれらのデータベースを用いる。

【 0 0 0 3 】このようなプログラム及びデータの族は、それらが全て1又は2以上のCPUからアクセスされている単一のアドレス空間 (AS) 内にあるとき、可能な最高速度で使用できることが分かった。しかしながら、その後の経験によれば、実行プログラムにより別の部分又はデータベースを消す間違った記憶動作により、前記プログラムの実行にかなりの障害があることが示された。前記実行の障害は一時的にこのようなシステムによる多重分岐バンキング業務の動作を停止した。全業務の一時的な停止によるプログラム・システムの障害は一般に、その動作速度に関係なく、許容できないものと考えられる。

【 0 0 0 4 】前記間違った記憶の重大さにかんがみ、トランザクションに代わって実行されるアプリケーション・コードから大部分のエラーが除去される保証を高めるために、より多くの日時及びCPU使用を含む更に厳重な検査方法を採用することが必要であった。

【 0 0 0 5 】前記プログラム障害を防ぐ明白な方法は、システム内の異なるプログラム及びデータベースを互いに分離することである。それによって、1つのプログラ

10

20

30

40

50

6

ムはそのシステム内の他のプログラム又はデータベースをアクセスできない。この分離は、予期された特定のトランザクションの範囲外の記憶が、それが試みられるとプロセッサによって捕捉され、間違った記憶が行われることを許さないように行われるべきである。そして、間違った記憶はオペレーティングシステムに報告され、オペレーティングシステムは、トランザクション・アプリケーションの族を制御する基本サブシステムに、そのトランザクションを放棄すべきことを通知する。エラー・メッセージにより不完全なプログラムの時宜を得た補正を行うことができる。このサブシステムは他の実行中のトランザクションを完全に制御する動作のままである。1つのトランザクションだけが回避され、それが本装置に通知され、間違った記憶が回避され、そしてともかく他のトランザクション又はデータは影響を受けずにすむ。

【 0 0 0 6 】仮想アドレスにより実行するプログラムの従来の保護手法は異なるアドレス空間内の異なるプログラム及びそれらのデータを分離することである。すなわち、1つのアドレス空間内で実行するプログラムは他のアドレス空間内のプログラム及びデータに影響を及ぼす間違った記憶を行うことはありえない。

【 0 0 0 7 】従来の技術は2つのタイプの仮想アドレス空間を開示している。その各々は仮想アドレスのシーケンスを表わしかつ分離特性を有する。それらは、第一に、プログラム・アドレス空間 (本明細書ではプログラム空間と呼ぶ) であり、第二に、データ・アドレス空間 (本明細書ではデータ空間と呼ぶ) である。2つのタイプのアドレス空間はコンピュータ・システムでASTE (ASN第2テーブル・エントリ) により指定される。

【 0 0 0 8 】各タイプのアドレス空間はコンピュータ・システムで変換テーブル指定 (TID) により与えられる。TIDは変換テーブル原点 (TTO) 及び変換テーブル長 (TTL) を含む。TTOはシステム・メイン・メモリ内の変換テーブルの位置を決め、TTLはそのエントリ数により変換テーブルの長さを指定する。2レベルの変換動作を用いるIBMの製品であるシステム/370及びESA/390構成では、TTO及びTTLを含むTIDはそれぞれSTO(セグメント・テーブル原点)及びSTL(セグメント・テーブル長)を含むSTD(セグメント・テーブル指定)として表示される。1レベルの変換動作を用いるコンピュータ構成では、TID、TTO及びTTLはそれぞれPTO(ページ・テーブル原点)及びPTL(ページ・テーブル長)を含むPTD(ページ・テーブル指定)により表示される。

【 0 0 0 9 】ESA/390構成では、各プログラム空間はそれに割当てられた特定のASN(アドレス空間番号)を有するのに対し、データ空間は割当てられたASNを有しない。プログラム空間は実行可能なコードを含み、かつデータを含むこともできる。データ空間はデータだけを含み、実行可能なコードを含まないが、実行不可能なコー

ドがデータとして含まれることがある。その構成及びオペレーティングシステムは命令がデータ空間で実行されないようにする。

【 0 0 1 0 】 非監視プログラムが別のプログラムを呼出すために1つのプログラム・アドレス空間から別のプログラム・アドレス空間に移行せねばならないときは、複雑な手順が必要とされる。1つの手法は、オペランドとしてPC番号を有する「ESA/390の"空間切替えによるプログラム呼出し"(PC-ss)命令を用いることである。PC番号は最初にPC番号変換を経てASN(アドレス空間番号)に変換される。次に、PC命令はASN変換を経てSID(セグメント・テーブル指定)に変換される。完全なPC命令の多重変換はシステム内のデータの完全性を保証するために多くの異なるタイプの権限検査を受ける。

【 0 0 1 1 】 PC-ssは米国特許出願第07/732936号(1991年7月19日出願)に開示されたものである。この特許出願は米国特許出願第07/154733号(1988年2月10日出願)に関連する。以上の2つの出願の内容は本発明の出願に参照される。

【 0 0 1 2 】 完全なPC命令はART(アクセス・レジスタ変換)を用いない。ARTはESA/390構成におけるデータ・アクセスにだけ用いられる。ARTプロセスはアクセス・レジスタ(AR)内のALETをSIDに変換する。

【 0 0 1 3 】 PC変換及びART変換の両者はASN第2テーブル・エントリ(ASIE)を用いる。ASIEの各々はコンピュータ・システムにより現に識別されたSIDを含む。ARTプロセスはそれがアクセスしたASIEを見つけるのにASNを用いない。ARTプロセスで用いるASIEは、もしそれがASNによって指定されなければ、"擬似ASIE"と呼ばれる。

これは、それがデータ・アドレス空間を表わす場合である。ASIEを指定するASNはASN変換にのみ使用される。これはプログラム呼出し(PC)及び関連命令のタイプ、並びにLASP(コード・アドレス空間パラメータ)命令により実行される。

【 0 0 1 4 】 本発明は、族グループのアドレス空間内のプログラム呼出しを行うとき、改良されたARTタイプの変換をASN変換の代わりに用いる。改良されたART変換プロセスは、そのグループ内の部分空間と基本空間の間の関係を利用することにより、完全なPC変換プロセスよりも複雑さがずっと少なく、完全なPC命令で用いられる大部分の権限検査を取り除く。それにもかかわらず、その族グループが最初に指名(dispatch)されるとき、又は制御がその族以外のプログラム空間に移されるかその族グループ以外のプログラム空間からその族に戻されるとき、なおPC権限検査が要求・依存・実行される。本発明は、グループ内でアドレス空間が切替えられる毎に指名プロセス後の全ての権限検査を反復することを不要にする。

【 0 0 1 5 】 記憶空間内のプログラムとデータを分離する別のある方法は、実即ち絶対主記憶空間で4KBブロックに

割当て可能な記憶キー保護を用いることである。使用できる記憶キーの数は(例えば、16個のキーに)限られている。異なる記憶キーに割当てられた記憶ブロック間の分離は、1つの記憶キーを有する1つの記憶ブロックから別の記憶キーを有する別の記憶ブロックへのキー切替えのために、問題状態プログラム要求を切替える監視(特権プログラム)制御を含む非常に大きいキー管理オーバーヘッドを必要とすることがある。(監視キー、キーのみが特定のPSWアクセス・キー管理を実行せずに他のキー領域をアクセスする。しかしながら、監視キーが使用されている間は異なるキー記憶領域間の分離機能が失われるので、監視キーは問題状態アプリケーション・プログラムにより使用することができない。)

【 0 0 1 6 】 例えば、もし非監視キーが第1及び第2のブロックに割当てられ、第2のブロック内のプログラムが第1のブロック内のデータをアクセスする必要があれば、第1のブロックに対するキーは一時的に第2のキー・ブロック内のプログラムに割当てられ、それが第1のキー・ブロックをアクセスすることを可能にする。従って、そのプログラムへのキーの割当て及び割当て解除を行うためにかなりの量のシステム・オーバーヘッドが必要となり、プログラムの動作への割込みと同時にオペレーティングシステム監視プログラムによるキー割当ての変更が必要となる。記憶キー割当て動作は保全上非常に注意を要するので、それらはメイン・オペレーティングシステム・プログラムによってのみ行われる。従って、記憶キー切替えは、異なるキーに割当てられたブロック間でプログラム切替え、即ち情報授受が殆ど又は全く必要としない場合にのみ利用できる。

【 0 0 1 7 】 プログラムは2以上のアクセス・キーの使用を許可されるが、ESA/390内の命令、例えばセットPSWキー・フロム・アドレス(SPKA)命令の使用により、PSWアクセス・キーの値の切替えを管理しなければならない。

【 0 0 1 8 】 キー保護は、関連したアクセス及び記憶キー、ならびにメモリ内の全ての4KBブロックのキーを記憶する特別記憶キー・ハードウェア・アレイの初期化を必要とする。プログラムのアクセス・キーはプログラムが実行されるPSW(プログラム状況ワード)内のフィールドにより指定される。記憶キーは実記憶の各ブロックと関連づけられる。キー保護は1つのプログラム内で用いられる複数のアクセス権限レベルを可能にする。このプログラムに2以上のキーをアクセスする能力を付与できる。この能力はPKM(プログラム・キー・マスク)と呼ばれる制御レジスタ・フィールドによって与えられ、実行されるタスクのオペレーティングシステムによって初期化される。しかしながら、一般に、1つの時点で1つのアクセス・キーの値しか使用できない。(ESA/390構成では、記憶領域内の1つのロケーションから別のロケーションにデータを移す小さなセットの命令が存在し、オ

9

ペランド毎に別個の保護キーを持つことを可能にする。1つのキーはPSWキーであり、他のキーは問題状態にあるPRMにより許可されるか、監視状態での実行により必然的に許可されなければならない。)

【0019】キー保護は同じアドレス空間又は別のアドレス空間で実行中のプログラムに適用することができる。従って、キー保護はアドレス空間保護を補完することができる。なぜなら、これらの異なる分離形式は異なる特性を有するからである。

【0020】共通記憶キー(PSK)の使用は、限定された環境の下にキー管理の必要性を取り除くことにより、プログラム及びデータの関連グループ(族)内のアドレス指定分離に対する部分的な解決を与える。共通記憶キーは、プログラムの一方向性アクセスを、PSKを割当てられた記憶ブロックに与える。即ち、PSKの使用は、PSK以外のキーに割当てられた記憶領域の全てのブロックを、PSKのアクセス権限の下に実行中のプログラムによる粗雑な記憶から保護することができると同時に、PSKを割当てられた記憶ブロックに対するキーに割当てられた全てのプログラムによる共通アクセスを可能にする。PSKは米国特許出願第07/710875号(1991年6月6日出願)に開示されている。

【0021】PSKを用いる利点は、一定の状況でキー管理プログラムの必要性を除去しうることである。キー一切替え制御なしに、PSKは監視キー又は任意の非監視キーを割当てられた記憶ブロック内のデータ及びプログラムを保護することができ、PSWでPSKモードの下に実行中のプログラムにより変更されないようにする。それらがアクセスする記憶領域はPSKを割当てられたブロック内にある。同時に、PSK記憶ブロックは、PSKの値を含む、任意のキーの値で実行中のプログラムにより変更できる。これは、族内の全てのプログラム及びデータをアクセスしかつ族の全ての部分に対するサービスを実行する基本プログラムを有するプログラム及びデータの族、とりわけ動作中の族及びオペレーティングシステムとの対話を必要とするものに特に有用であるが、非基本プログラムにより行われる粗雑な記憶から保護されることを必要とする基本プログラムがある。

【0022】例えば、保護キーのセット0-15により、16個の保護キー全部について、共通記憶キーはキーの値9を、監視キーはキーの値0を、そして非監視キーはキーの値1-8及び10-15を持つことができる。また、記憶保護キー0-15の各々は関連した取り出し保護フラグ・ビット(FP)、基準ビット(R)及び変更ビット(C)を持つことができる。保護キーはどれもプログラムのPSW(プログラム状態ワード)に配置されたアクセス・キーとして、又はコンピュータの主記憶領域内のそれぞれの4KBブロックに関連したエントリを有する記憶アレイに配置された記憶保護キーとしてそのブロックの保護キーとともに用いることができる。

10

【0023】動作規則は、任意の非監視キー(1-8及び10-15)のユーザが任意の共通キー(9)の記憶ブロックをアクセスできることである。しかし、共通キー(9)のユーザは、監視キー(キー0)又は非監視記憶キー(キー1-8及び10-15)により保護された記憶ブロックに記憶する(又はもし取り出し保護が活動状態であれば取出す)ことができない。

【0024】従って、共通アクセス・キー9により実行中のプログラムは、記憶キー内のFPビットの設定に関係なく、PSK9を有するブロックにだけ記憶できる。しかし、共通アクセス・キー9を有するプログラムによる取り出しはFPビットの設定により制御される。もし記憶キーのFPが1であれば、一致が存在する場合にのみ取り出しが可能である(キー9のプログラムを、PSK9を割当てられた記憶ブロックだけのアクセスに制限する)。しかし、もし記憶キーのFPが0であれば、一致がなくとも取り出しは可能である(共通キー9のプログラムは任意の記憶キーに割当てられた記憶ブロックをアクセスできる)。

【0025】任意の非監視キー(1-8及び10-15)又は監視キー0を割当てられたプログラムは、PSKブロックのFPビットの状態に関係なく、一致なしに任意のPSKブロック(キー9)に記憶又取り出すことができる。

【0026】それにもかかわらず、プログラムの最高実行速度を得るために(たとえPC及び関連した命令並びに記憶保護キーが使用可能であっても)分離保護なしに1つのアドレス空間で実行する使用頻度の高いプログラムの族が存在する。前記プログラムでプログラムとデータの間の分離の欠如によりデータの完全性の破局的な障害がときどき起きる。これは前記プログラム及びデータによる連続的で中断されない動作により業務に重大な対価を強いことがある。更に、データの完全性の侵害が起きると、前記侵害は一般にその発生の瞬間に発見されないが、後に気付かれるか、又は将来のある時点で間違った結果を生じたときに発見される。

【0027】PSKにより分離要求の部分をアドレス指定する手段が提供された。これはPSKの制御の下に実行され、それらの制御によりサービスされるアプリケーション・プログラムと基本サブシステムの間の分離を可能にした。本発明は、各々の1つを他の全てから保護する、アプリケーション・コード・モジュールとデータの分離に拡張される。実行中の各トランザクションを他の全てのトランザクションから分離する仮想アドレス指定能力を用いる本発明の出願で、PSKは、サブシステム自身を全てのトランザクションから保護することにより全システム分離の一部に留まると同時に、それによって、そのサービスを実行する際の全てのトランザクションのコード及びデータへの高性能の直接アクセスを可能にする。

【0028】

【発明が解決しようとする課題】本発明の主たる目的は

50

コンピュータ・システムがデータ及びプログラムの完全性を維持すると同時に関連プログラム及びデータのグループ(族)の高速実行を可能にする方法及び手段を提供することにある。特に、粗雑な記憶はプログラムを破壊しデータを無効にするから、それぞれのプログラム及びデータを含む記憶領域は他のプログラムの粗雑な記憶から保護する必要がある。高速実行の目標は、プログラム族が1つのアドレス空間内に含まれ、族内のプログラムとデータの間の粗雑な記憶から分離されない場合の取得可能な速度に近づくことである。

【 0 0 2 9 】

【課題を解決するための手段】そのために、本発明はグループ内のプログラムを指名するCPUに対するグループの識別子を設ける。グループ識別子はグループ指名が活動状態である限り変らない。グループ識別子は指名するCPU内の制御レジスタで見つかるか、又はその識別子を含むメモリ制御ブロックを指すアドレスとことができる。

【 0 0 3 0 】本発明は(前述の既存のプログラム空間及びデータ空間の外に)新しい2つのタイプの仮想アドレス空間を用いる。これらの各アドレス空間はそれぞれのTIDにより与えられる。新しいアドレス空間は(1)実行可能なコードを含むプログラム・アドレス空間の特別なタイプであるが割当てられたASNを持たない部分アドレス空間(本明細書では、部分空間と呼ぶ)及び(2)追加として部分空間に関連するプログラム・アドレス空間である基本アドレス空間(本明細書では、基本空間と呼ぶ)である。基本空間は任意の数の部分空間及び少なくとも1つの部分空間を持ちうる。基本空間はそのTIDにより与えられる。新しいフラグ表示は、新しい2つのタイプのアドレス空間を識別するTIDで与えられる。

【 0 0 3 1 】各基本空間は特定のASNを割当てられるが、部分空間は特定のASNを割当てられない。しかし、各部分空間はそれが関連基本空間のASNを有するとみなされる。基本空間及びそれが関連した全ての部分空間はまとめてアドレス空間グループ、即ち族(family)とみなされる。基本空間のASNはグループ全体を表わす。

【 0 0 3 2 】プログラム実行の速度は、同じアドレス空間内のプログラム間の制御移行により空間間(inter-cpa ce)分岐のオーバーヘッドなしに通常の分岐を用いることができるので減速されない。1つのアドレス空間内のプログラムから別のアドレス空間内のプログラムに実行を切替え、前のプログラムの権限及びアドレス指定状態を補完し、そして次のアドレス空間内のプログラムが必要とする権限及びアドレス指定状態を確立することにより、実行が減速される。プログラム制御の切替え中に両アドレス空間内のデータ及びプログラムの完全性を維持するために空間間の制御が実行されるので、故意の又は間違った呼出し元は次のアドレス空間内のプログラム又はデータを誤ってアクセスする権限を得ることはなく、

その逆も起こりえない。

【 0 0 3 3 】アドレス空間切替えの速度は、本発明により提供された、"部分空間グループ内の分岐" (BSG) と呼ばれる新しい問題状態分岐命令により、従来のアドレス空間切替え方法よりも大幅に改善される。この命令は、基本空間及び複数の部分空間を含む異なるグループ(族)のアドレス空間内のプログラム間でCPU実行を切替えるように動作する。

【 0 0 3 4 】BSG命令は、基本空間内のプログラムから部分空間内のプログラムへの分岐、部分空間内のプログラムから基本空間内のプログラムへの分岐、及び部分空間内のプログラムから別の部分空間内のプログラムへの分岐を行うためのグループ内(intra-group)プログラム呼出しである。

【 0 0 3 5 】本発明のグループ内プログラム呼出しのアプローチは、部分空間/基本空間の間のCPU制御を切替えるために完全なプログラム呼出し(PC)命令を頻繁に使用するためのオーバーヘッドの欠点の大部分を除去できる。これは、部分空間グループ内で実行する仮定が正規のPCインターフェースの検査制御を必要としないために、又は新しい情報がプロセッサ制御レジスタをセットする必要がないように制御移行の両サイドに同じ動作状態を与えるために可能である。特定の例として、拡張権限インデックス(EAX)、PKM、PSWキー及び問題状態は変わらない。これらの権限制御に関する情報を保管する連係スタック、即ちGPR又はARの値は必要ではない。後者については、レジスタ保管及び復元はプログラム自身によるプログラム連係の際に処理しなければならない。前者については、族のメンバー間に要求される分離は記憶領域のアドレス指定能力を制約するものであり、エラーによる記憶を阻止・報告し、前記エラーの影響をそのエラーを含むプログラムに抑えるようにする。基本的な権限特性は族のコード内で管理された一定の変更により族を通じて同じに維持される。この例はPSWキーであり、族のPKMにより強制されたときSPKAにより変更される。本発明はグループ内で用いる完全なPC命令を回避し、完全なPC命令を用いる必要性を、グループ外のプログラム空間に対して行われるプログラム呼出しに限定する。この場合、通常の空間間分離が必要である。

【 0 0 3 6 】部分空間はそれ自身の特定のASNを持たないが、基本空間のASNの要求に従って表示される。それについては後に詳細に説明する。グループ外のアドレス空間からその部分空間グループへのPC命令の場合に、そのPCの目標ASNは基本空間内のプログラム・エントリ・ポイントを要求する。この場合、前記ポイントはASNを表わす。

【 0 0 3 7 】従って、本発明はアドレス空間の"グループ内"でプログラム制御を移動する方法を提供する。この場合、基本空間はこのグループ内の部分空間のサーバー・アドレス空間として動作する。しかしながら、グル

ープ内の部分空間にあるプログラムは、グループ外のアドレス空間に対するプログラム呼出しを行うことができるが、簡略化されたプログラム呼出し、部分空間グループ内の分岐(BSG)はグループ内でしか行うことができないから、このような呼出しは完全なPC命令でなければならない。PCが用いられると、プログラム復帰(PR)命令は呼出し元の部分空間に制御を戻す。

【0038】最初に、前記グループ内のプログラムはアドレス空間のグループの指名で開始される。前記指名はグループ内の任意のプログラムの実行開始前にASN変換を実行するLASP命令を用いる。ロードPSW命令は指名されたプログラムの権限状態を設定する。そして、開始されたプログラムは、グループ内の基本空間内又は任意の部分空間内のプログラムを呼出すために、BSG(部分空間内の分岐)命令を使用できる。BSG命令は、部分空間グループ外のプログラムの呼出しに使用できない、グループ内の空間横断分岐命令である。

【0039】BSG命令は、1つの族グループ内のアドレス空間の基本空間／部分空間にあるプログラム間で分岐するためにだけ用いるように、その使用を制限する機構を用いる。

【0040】本発明は、現在のソフトウェア構成の最小の変更により、基本空間に従属するアドレス空間として部分空間を実現できる。それはサブシステム・アプリケーション構成を可能にする。この構成では、サブシステムは全てのプログラムをアドレス指定し、そのサブシステムによりこれらのプログラムに割当てられたトランザクションを実行できるが、そのサブシステムからのサービスを得るためにプログラミングで指定された正規のインターフェースを介して指定されたエントリ・ポイントに制御を移し又は異常の発生を報告する能力の外は、これらのプログラムの各々のアドレス指定能力をそれ自身の記憶領域に制限する。IBM CICSサブシステムはこのような構成の一例であり、変換アプリケーション・コードからの記憶からそれ自身を保護する従来のPSKの発明を使用し、そしてこの発明はトランザクション・アプリケーション・プログラムの間のアドレス分離を与えるために用いられる。

【0041】ESA/390環境では、BSG命令実行はCPU制御レジスタ内のアドレスに位置するメモリ・ブロックにあるDUCT(指名可能装置制御テーブル)内の情報の状態を用いる。本発明はグループ内の基本空間及び最後に加えられた部分空間の識別子を記憶し、グループ内の現在のプログラム又は最後に実行されたプログラムが基本空間内にあるか部分空間内にあるかを示すフィールドをDUCT内に設ける。グループ内のプログラムの実行が割込まれた場合に、オペレーティングシステム(OS)が(その次の指名が開始するか又はOSが割込み処理後に制御を戻すときのように)そのプログラムの実行を再開始するとき、この情報が使用されて前記割込まれたプログラムに制御を

戻すことができる。同様に、グループ内のPC命令の使用によりグループ外のプログラムが呼出されると、DUCTは制御が基本空間又は部分空間のどちらからグループを去ったかに関する情報を保持する。それはもし部分空間からであれば1である。グループ内の呼出し元のプログラムに戻す次のPRの際に、グループ内の正しい空間に実行制御を取得するためにDUCTが用いられる。

【0042】各部分空間は変換テーブルを指定する特定のSTDを有する。変換テーブルはそれぞれの部分空間又は基本空間内の仮想アドレスの変換から得られる実アドレスを与える。本発明の実施例は各部分空間の一部も基本空間でアドレス可能であることを示すが、これは本発明では要求されない。部分空間が設定されると、基本空間と部分空間グループの部分空間のどれか又は全てとの間の共通のアドレス指定能力なしに、BSGを用い基本空間及びその部分空間の間で実行制御を移すことができる。

【0043】本発明はグループ内の実行中の部分空間又は基本空間がともにその現在の一次及び二次アドレス空間としてCPUに登録される概念を与える。これは、別のアドレス空間内の呼出し元プログラムをアクセスするための情報をトランザクションに与えずに変換分離を援助する。従って、BSG命令が呼出された部分空間に切替えられると、BSG実行はCPUの第1及び第2のSTDフィールド(ESA/390 CPUのCR1及びCR7にある)をどちらもセットする。これは、呼出されたプログラムにより CR7がその呼出し元の記憶領域をアクセスするのに用いられるのを阻止する。部分空間はBSGだけを用いて、そのアクセス・リストの1つにより明白なアドレス可能性を有する、(グループの内部の)アドレス空間にCPU実行制御を移すことができる。BSG命令と異なり、(従来のアドレス空間の間のプログラム呼出しとして用いられる)完全なPC命令を指定して、CR7内の呼出し元アドレス空間のSTDを第2のSTDとして保管することができる。第2のSTDは呼出されたプログラムによって用いられ、別のアドレス空間内のその呼出し元プログラムをアクセスできる。

【0044】要約すると、本発明の目的は、基本空間とその関連部分空間の間に存在する所定の権限関係に依存するグループ内プログラム呼出し命令を提供し、完全なプログラム呼出し命令によって行われる権限検査及び他の機能の大部分を除去することにある。代わりに、本発明は能率化された(streamlined)アクセス・レジスタ変換を用いてグループ内プログラム呼出しを実行する。グループ内プログラム呼出しは、そのグループを指名するプログラムを指名するオペレーティングシステムにより最初に行われる権限検査に依存する。そして、グループ内のプログラム呼出しのためのこれらの権限検査の反復は本発明により除去される。

【0045】

【実施例】図1及び図2は本発明で用いられる仮想アドレ

ス変換テーブル間の関係の例を表わす。これらの図面の各々で、それぞれのアドレス空間のTID(変換テーブル記述子)をアクセスすることにより最高のレベルが得られる。

【 0 0 4 6 】 [ BS及びSSの2 レベル仮想アドレス指定 (図1) ] 図1に示す各TID内のTIOはST0(セグメント・テーブル原点)である。ST0はそれぞれのアドレス空間、基本空間(BS)又は部分空間(SS)のメイン・メモリ内のセグメント・テーブル(ST)のアドレスである。2 レベル変換動作は良好な実施例で用いられる。

【 0 0 4 7 】 ESA/390及びS/370構成では、図1に示すように、STは最高レベルの2 レベル仮想アドレス変換テーブルである。各STはそれぞれのアドレス空間として指定された全てのセグメントのエントリ(STE)を含む。図1で、基本空間STはB-ST 20と呼ばれ、各部分空間STはS-STと呼ばれる。S-STはS-ST1 21及びS-ST2 22の2つが示されている。

【 0 0 4 8 】 STの各々は、それぞれアドレス空間のセグメントを定める、メモリ内の対応するページ・テーブル(PT)を見つけるセグメント・テーブル・エントリ(STE)を含む。STEの各々はそれぞれのページ・テーブル(PT)をアドレス指定するページ・テーブル原点(PTO)を含む。これはアドレス空間の当該セグメントのアドレス指定能力を表わす。図1に示す基本空間は、B-ST 20内の6つのSTEによりアドレス指定される6つのPT(PT1-PT3及びPT5-PT7)を有する。

【 0 0 4 9 】 図示のように、第1の部分空間がS-ST1 21で表示される。これはPT2、PT3及びPT4の3つのPTを見つける。セグメント・テーブル S-ST2 22で第2の部分空間が表示される。これはPT4、PT5、PT6、PT7及びPT8の5つのPTを見つける。基本空間20にだけ用いるPT1は部分空間の一部ではない。第2の部分空間22が有するPT8は基本空間又は他の部分空間とは共有されない。更に、PT4は部分空間 S-ST1 21及びS-ST2 22と共有されるが、基本空間B-ST 20とは共有されない。

【 0 0 5 0 】 PTの各々は複数のページ・テーブル・エントリ(PTE)を有する。PTEの各々は、コンピュータ・システムの実メモリ内のそれぞれの4 KBページ・フレームを見つけるアドレスを有する。それぞれのアドレス空間のSIDによりアドレス指定できる全てのPTEは、それぞれのアドレス空間内でアドレス指定できるシステム・メモリの部分を決める。通常、これは実メモリ全体よりも小さい。

【 0 0 5 1 】 従って、図1は、本発明で用いられる、関連共有(又は非共有)実アドレス指定が仮想アドレス変換テーブル(ST及びPT)により取得される方法を示す。

【 0 0 5 2 】 [ BS及びSSの1 レベル仮想アドレス指定 (図2) ] 図2は1 レベル仮想アドレス変換テーブル構成によって得られた関連共有(非共有)アドレス指定を示す。これは本発明で用いることもできる。図2に示す基

本空間はB-PT 30内の6つのPTEによってアドレス指定される6つのページ・フレーム(PF1-PF3及びPF5-PF7)を有する。PF1は部分空間の一部ではない。

【 0 0 5 3 】 図示のように、第1の部分空間がS-PT1 31で表示される。これはPF2、PF3及びPF4の3つのPFを見つける。ページ・テーブルS-PT2 32で第2の部分空間が表示される。これはPF4、PF5、PF6、PF7及びPF8の5つのPFを見つける。PF5～PF7は基本空間と共有されるが、PF8は基本空間とは共有されない。更に、PF4は2つの部分空間S-PT1 31及びS-PT2 32とで共有されるが、基本空間B-PT 30とは共有されない。

【 0 0 5 4 】 図1と同様に、図2では、各PTは複数のページ・テーブル・エントリ(PTE)を含み、各PTEはPFRA(ページ・フレーム実アドレス)と呼ばれるアドレスを有する。これはコンピュータ・システムの実メモリ内のそれぞれの4 KBページ・フレームを見つける。図2でそれぞれのアドレス空間のPTOによりアドレス指定できる全てのPTEは、それぞれのアドレス空間によってアドレス指定できるシステム・メモリの部分を決める。通常、これは実メモリ全体よりも小さい。

【 0 0 5 5 】 従って、図2は、本発明でも用いられる、1 レベルの仮想アドレス変換テーブル(PTのみ)により取得される関連共有(又は非共有)実アドレス指定を示す。

【 0 0 5 6 】 [ 基本空間と部分空間の間の共通アドレス指定 (図3、図4及び図5) ] 図3は本発明をMVS/ESAのようなオペレーティングシステムで用いる方法の例である。部分空間グループ内のアドレス空間は列(column)40、41及び42で表示され、列40は基本空間を表わし、列41及び42は部分空間グループ内の部分空間を表わす。列43は部分空間グループの一部ではない。この場合のオペレーティングシステム(OS)の制御プログラムは、それがサポートする全てのアドレス空間に写像される。この例では、部分空間グループは、部分空間に作業を指し、プログラムされたサービスを部分空間グループで実行中のプログラムに与える共有サブシステム(サブシステムA)の周りに構築される。サブシステムAは基本空間及び全ての部分空間にアドレス写像されるが、グループ外のアドレス空間には行われない。

【 0 0 5 7 】 図4は、図3に示されたプログラミング構成のアドレス指定能力が、ESA/390アドレス指定の際に、図1に示した共有ページ・テーブルの方法で達成される方法を示す。図3の基本空間40及び部分空間の1つ(41)のアドレス指定構成が40A及び41Aに示される。アドレス空間の各々はそれ自身が定めるSIDを有し、共有セグメントについては、同じページ・テーブルの各アドレスに適切なセグメント・テーブル・エントリを有する。

【 0 0 5 8 】 図5は図3のアドレス指定能力が図2に示したような個々のページ単位で達成される方法の例である。ここでは、ページ・テーブル40C及び41Cは共有でき

ないが、個々の仮想ページは両アドレス空間内のアドレスを同じページ・フレームに変換させることができる。共有の細分性の可能性は全セグメントから単一のページに減少する。

【 0 0 5 9 】 [ 基本空間と部分空間の関係 ( 図 6 ) ] 図 6 で、グループ内のアドレス空間は列 40-43 で表示され、列 40 は基本空間を表わし、列 41, 42 及び 43 はそれぞれ部分空間 SS1, SS2 及び SS3 を表わす。この良好な実施例により 2 レベル変換テーブルが用いられる。よって、基本空間 40 はセグメント・テーブルを有し、部分空間 41, 42 及び 43 の各々はそれぞれセグメント・テーブルを有する。基本空間及び各部分空間のセグメント・テーブル指定はそのそれぞれの ASTE 内にある。

【 0 0 6 0 】 基本空間 40 は、それぞれのページ・テーブル PT1, PT2, PT3 及び PT4 によってアドレス指定されたプログラム C1, T1, T2 及び T3 を含む。PT1 内のプログラム C1 は全てのアドレス空間 40, 41, 42 及び 43 に共有される。PT2 内のプログラム T1 は基本空間 40 及び部分空間 41 にのみ共有される。プログラム T2 は基本空間 40 及び部分空間 42 にのみ共有される。そして、プログラム T3 は基本空間 40 及び部分空間 43 にのみ共有される。共通にアドレス指定するために、PT は適切な ST 内の有効な STE によってアドレス指定されなければならない。( 本発明の原理を説明する図面を簡略化するために図 3 には 4 つのプログラムが示される。これは、実際のシステムを特許図面に示すことはできず、数十、数百又は数千のプログラム及びデータ領域が 1 つの部分空間に存在し、その基本空間に反映されるからである。 )

【 0 0 6 1 】 アドレス空間 40-43 の構成は、1 つのアドレス空間内で実行中のプログラムが粗雑な記憶を他のアドレス空間に送り又は他のアドレス空間から受取ることを、共通にアクセスできるアドレス空間の領域を除いて阻止する。例えば、部分空間 42 は記憶ブロック C1 及び T2 のみをアクセスし部分空間 42 は記憶ブロック T1 及び T3 をアクセスできないので、ブロック T1 及び T3 は部分空間 42 で実行中のブロック T2 からの粗雑な記憶から保護される。ブロック T1 及び T2 は部分空間 43 内のブロック T3 からの粗雑な記憶から保護される。同様に、T2 及び T3 は部分空間 41 内で実行中の T1 から保護される。

【 0 0 6 2 】 基本空間 40 の中の C1, T1, T2 及び T3 領域内のブロックは関連部分空間 41, 42 及び 43 内の対応するブロックと共通にアドレス指定可能である。よって、基本空間及び各部分空間はそれらが共有するブロック内のプログラム及びデータのみをアクセスできる。それらは、図 1 で説明したように、それぞれのアドレス空間のセグメント変換テーブルによりアドレス可能なブロックである。

【 0 0 6 3 】 本発明のもう 1 つの実施例では、部分空間グループのアドレス空間の間の共通のアドレス指定能力を必要とせずに、その部分空間グループを設定すること

ができる。

【 0 0 6 4 】 [ 共通記憶キーの使用による新たな保護 ] 本発明は、粗雑な書込みに対する最も有効な保護を得るために、共通記憶キー・タイプの記憶保護がグループ・アドレス空間タイプの記憶保護と組合わされることを可能にする。これは本発明の良好な実施例で示される。共通記憶キー保護は、そのキー保護が基本空間又は任意の部分空間を含むアドレス空間内で動作すると同時に、アドレス空間保護がアドレス空間同志の間で動作するので、アドレス空間保護を補完する。従って、各部分空間の特定の内容が PSK に割当られると同時に、一部の又は全てが共有するコード又はデータは異なる記憶キーに割当ることができる。

【 0 0 6 5 】 実メモリには 1 つの実際の記憶ブロック C1 だけがある。ブロック C1 はグループ内の全てのアドレス空間の全セグメント・テーブル内にないので、グループ内の全てのアドレス空間で共通に用いられる。C1 は PSK に割当られないことにより保護される。

【 0 0 6 6 】 もし、C1 が、全ての部分空間に写像される、1 つのトランザクション・コードよりも多くの情報を含むデータ・バッファを含む、サブシステム・サービスの共有セットを含むならば、そのトランザクションは PSK に割当られ実行することができる。共有のコード及びデータ・バッファは別の記憶キーを割当できるので、全てのトランザクションからの記憶から保護することができる。しかしながら、トランザクション及びそれらの専用データ領域が PSK を割当てられるから、サブシステム・サービスは、特定のトランザクション領域への取出し即ち記憶を必要とする所要のサービスの動作で、これらを直にアクセスできる。

【 0 0 6 7 】 PSK 割当ては、ブロック T1, T2 及び T3 が PSK を割当てられている同じ部分空間内の他の記憶ブロックのどれかに、PSK ブロック内のプログラムが記憶させることを阻止する。そして、任意の T プログラムにより生成された粗雑な記憶は、同じ部分空間、例えばサブシステム C 領域の中の別の保護キーを割当てられたブロックに到達できない。しかし、( PSK が割当てられた ) T ブロックは同じ部分空間内で別のキーを割当てられる記憶ブロック内の全てのプログラムによりアクセスできる。従って、ブロック T1, T2 及び T3 は異なる保護キーを割当てられた他のブロックの粗雑な記憶からは保護されないが、これらは相対的に安定しかつ十分に検査された主要なサービス・サブシステムのモジュールであると想定される。しかしながら、部分空間構成は T 領域を相互の粗雑な記憶から保護する。これは、部分空間構成で、それらの相互のアドレス指定能力を拒絶することにより達成される。まとめとして、共通サブシステム・サービス提供者及び共有サービスを用いる複数のトランザクションの構成で、サブシステムが異なる保護及びアクセス・キーを有する間、トランザクション記憶領域に対するサー

ビス・コードの直接のアドレス指定能力を失わずに、トランザクション領域は部分空間の限定されたアドレス指定能力により保護されると同時に、全てのサブシステムでアドレス可能なように写像された共有サービス・サブシステムは、その保護及びアクセス・キーとして各トランザクション領域にPSKを割当てることにより保護される。

【0068】[関連アドレス空間の変換テーブル指定]  
本発明の良好な実施例で、TIDはSID(セグメント・テーブル指定)である。これは図6でASTE 71内のワード73として示される。STO(セグメント・テーブル原点)はSID内のフィールドである。これはコンピュータの主記憶装置内のセグメント変換テーブルの実アドレスを含む。そして、SILはセグメント変換テーブルの長さを表わすSTD内のフィールドである。

【0069】STDは、STDがアドレス空間のグループ内のアドレス空間を決めるかどうかを示す新たなフラグ・フィールド(Gビット)も含む。もしG=1であれば、SIDは部分空間グループ内のアドレス空間を与える。もしG=0であれば、そのアドレス空間は部分空間グループにはない。

【0070】もしフィールドGがSIDのグループの存在を表わせば、SIDを含むASTEはフラグ・フィールドB(図6でASTE 71内のフィールド72)も有する。これはSIDがグループ・アドレス空間の基本空間を表わすか部分空間タイプを表わすかを示す。もしB=1であれば、SIDは基本空間を表わし、もしB=0であれば、SIDは部分空間を表わす。

【0071】[同じ部分空間グループの異なるアドレス空間内のプログラムの間の分岐]実行中のプログラムはBSG命令の使用により関連部分空間内の同一の又は他のアドレス空間内の命令に分岐できる。BSG命令はそれが実行している1つの部分空間グループ内で動作するように制限される。アドレス指定能力を除いて、部分空間内で実行中のタスクは、前記属性、例えばPSWアクセス・キーを変更するために、指定された構成を用いるプログラム・ステージをそのタスクが用いない限り、任意の一時点でそれがどの部分空間のグループに入っているかに関係なく共通の権限属性を有する。また、ESA/390の他の分岐として動作する命令が与えられ、レジスタ(GR及びAR)の保管及び復元は連係に必要なプログラムの責任である。これはこれらの連係での動作の局所的な最適化を可能にする。部分空間グループは全体に統一された構内通信の設計による関連プログラムのセットを含むと仮定される。他方、ESA/390(PC, PR, PT)の正規の機構は、きびしい分離が要求されることと、主要なプログラミング・サブシステムがそれらのサービス呼出し側又は被呼出し側からの完全な独立を望むことを仮定する。従って、それらはより精巧な権限検査を課し、かつ(又は)ずっと多くの構成情報を保管・復元する。例えば、ESA/

390 PC命令は、別のアドレス空間への分岐を実行するプロセス中に、BSG命令よりもずっと多くのメモリ・アクセスを行う。

【0072】本発明のBSG命令の本質は、部分空間グループ内の空間切替えを伴うプログラム分岐のための省略タイプのアクセス・レジスタ変換(ART)プロセスの使用である。空間分割(例えば、PC/PT/PR)を伴う従来の分岐命令はARTプロセスを用いなかったが、PC番号変換とそれに続くASN変換を使用した。前記変換はともにARTプロセスにはなかった多くの許可検査を含んでいた。ARTプロセスは以前にはプログラム分岐制御では使用されなかった。すなわち、ARTプロセスは以前はプログラム実行中に異なるアドレス空間内でデータ(オペランド)をアクセスするためにのみ使用された。

【0073】本発明は、1つのアドレス空間内のプログラムが別のアドレス空間内の他のプログラムの呼出しを必要とし、制御移行が部分空間グループ内の関連アドレス空間で行われるとき、それらの相互関係が、呼出されたプログラムを実行させる呼出しプログラムの権限検査を緩めずに、空間切替えを伴う従来の分岐命令(例えば、PC-ss命令)で用いる検査の大部分の除去を可能にすることが分かっている。これらの検査は前述の米国特許出願第07/732936号に開示されている。

【0074】[部分空間グループ内の分岐(BSG)命令で用いる資源]

部分空間グループ内の分岐(BSG)命令は簡単な形式: BSG R1, R2を有する。ここで、BSGは命令コードであり、R1及びR2はBSG命令を実行する際に用いられるオペランドを含むCPU内の汎用レジスタである。汎用レジスタR2は宛先アドレスを含み、R2に関連したアクセス・レジスタ(AR2)は宛先アドレス空間を表わすALETを含む。R1は、R2アドレスでプログラム呼出しが終了した後の戻り点を与える、BSG命令に続く命令のアドレスを受取る。

【0075】図6は良好な実施例でCPUがBSG命令を実行する際に用いられる重要なエレメントの一部を表わす。例えば、汎用レジスタR1及びR2はESA/390中央プロセッサ内の16個の汎用レジスタのどれかであり、アクセス・レジスタ(AR)は16個の汎用レジスタの各々に関連する。レジスタR2に関連したARはAR2と呼ばれる。もしR2が0に指定されれば、AR2の内容はALET 0であるとみなされる。図6のR1 61及びR2 62を参照されたい。R1に関連したAR1はBSG命令では用いられない。AR2はALET(アクセス・リスト・エントリ・トークン)を含む。これは従来の技術では、Pビット、ALESN(アクセス・リスト・エントリ・シーケンス番号)及びALEN(アクセス・リスト・エントリ番号)を含む。良好な実施例ではPビット及びALENのみがBSGプロセスで用いられ、ALESNは用いられない。図6はBSGが制御を空間40から空間41に移すために初期化されたAR2/R2を示す。BSG命令実行の

21

一部として R1 60が、図示のように、BSGに続く命令の仮想アドレスを受取る。

【 0 0 7 6 】 指名中のCPU内のグループの識別子の記憶を含む、部分空間グループのアドレス空間の定義は、BSG命令を実行できるグループ内で実行中のタスクの前提条件である。ESA/390システムでは、プログラム・タスクは(MVSのような)ソフトウェアのオペレーティングシステムにより指名された作業の単位である。タスク制御ブロック(TCB) 50はコンピュータ・システムにより実行されるタスク毎に与えられる。IBM MVS TCBが図6に示されている。TCB及び幾つかの関連制御ブロックがオペレーティングシステム・ソフトウェアにより生成された後に、指名を行うことができる。オペレーティングシステムにより最初にメイン・メモリに生成された他の制御ブロックは、図6に示す指名可能装置毎の DUCT (指名可能装置制御テーブル) 及び DU-AL (指名可能装置アクセス・リスト) を含む。

【 0 0 7 7 】 指名タスクのプロセス中に、主記憶領域内の DUCT を見つけるために、タスクの DUCT ポインタ (PTR 52) が指名 CPU の制御レジスタ 2 (CR2) にロードされる。DU CT はコンピュータ・システムのメイン・メモリ内の DU-AL 66 を見つけるポインタ (これは実アドレスである) を有する DUAL (指名可能装置アクセス・リスト 指定) を含む。DU-AL は、グループ内の全ての部分空間 (SS1, SS2 及び SS3) をそれぞれ表わす ASTE を見つける実アドレスを含む ALエントリ (ALE) のセットを含む。現在の BSG命令の AR 2 62 内の ALEN により特定の部分空間が得られる。これは、BSG プログラム呼出しの宛先部分空間を指定する ASTE 71 をアドレス指定する (例えば SS1 の) 宛先 ALE を選択する。この宛先 ASTE は宛先 STD を含む STD フィールドを含む。これは BSG命令により開始される部分空間を決める。AR2 内の P ビットを用いて前記プロセスを DU-AL の代わりに PS-AL に振り向けることができる。

【 0 0 7 8 】 変更された APT プロセスは、指定された ALE 71 を BSG の第2 のオペランドの部分として変換するために用いられる。ALE 0 及び 1 は下記の特別の意味を有する。ALE 71 の ALESN と ALE とは比較されない。それらは無視される。ALE を用いるときの許可は検査されない、すなわち CPU 制御レジスタ内の ALE 及び EAX にある ALEX、ALE 71 の専用ビット、及び指定 ASTE から指定された ESA/390 権限テーブルは全て無視される。

【 0 0 7 9 】 R1 59 は PSW ビット 32-63 の内容 (アドレス指定モード及び命令アドレス) を受取り、CR1 は宛先 ASTE 71 から得られた STD によりロードされる。BSG の R2 のビット 32-63 は PSW ビット 32-63 にロードされる。

【 0 0 8 0 】 CR、AL、その ALE 及び ASTE は、ESA/390 Principle of Operation (form number SA22-7201-00) にあるような、形式及び方法が従来の技術で決められる。

【 0 0 8 1 】 図6 は本発明により DUCT を変更する方法を示し、図示のように新しいフィールド 55、56、57 及び 58

10

20

30

40

50

22

を含む。これらのフィールドはそれぞれ次の内容: SA (部分空間活動状態ビット)、SSASTEO (部分空間 ASTE 原点)、現在の SSASTEO の値が DUCT に置かれたときの SSAST ESN (部分空間 ASTE シーケンス番号) 及び BASTEO (基本空間 ASTE 原点) を有する。SSASTESN は予想された SSASTE が変更されなかつたことを検査するために用いられるシーケンス番号であり、オペレーティングシステムにより部分空間 ASTE 内の ASTESN の値を変更するだけで関連アドレス空間に対する権限のアクセスを取り下げることができる。BASTEO は現に指名されたタスクの基本空間を識別する。そして、部分空間グループ動作中に基本空間が変わらないので、タスクが CPU に指名されている限り BASTEO は変わらない。SSASTEO はタスクが実行中であったか又は実行されつつある最後の部分空間を識別する。タスクが部分空間で実行されるとき SA ビットは 1 にセットされ、タスクが基本空間で実行されるとき SA は 0 にセットされる。この情報は、最後の指名の終りでそのタスクが実行中であったアドレス空間を、各次のタスクの指名が決定することを可能にする。最後の指名の割込みからの古いプログラムの PSW は最後の実行プログラム内の最後のアドレスを含むので、各次の指名は最後の指名が終了した命令からプログラム実行を継続できる。DUCT は DUCTO (DU CT 原点) を含む制御レジスタの内容によりアドレス指定される。制御レジスタ内の DUCTO は指名可能な装置又はタスクが CPU にされている間は変わらない。

【 0 0 8 2 】 図7 は与えられたグループ構成部分の標識による ESA タイプのセグメント・テーブル指定を示す。

【 0 0 8 3 】 [ BSG 命令の動作 (図8) ] 図8 は BSG 命令の動作を表わす流れ図である。

【 0 0 8 4 】 ステップ 81 は、BSG が部分空間内で実行中であることを、現在の PASTEO を DUCT の BASTEO と比較することにより検査する。それらは、部分空間グループの基本空間又は任意の部分空間で実行中は等しいが、例えば、PC が部分空間グループ外のプログラムを実行するために用いられ、かつ実行制御がそのグループに戻っていない場合には等しくない。もしそれらが等しくなければ、BSG は許可されず、ステップ 82 で例外が通知される。BSG 動作で ALET 0 は部分空間から基本空間に復帰される。BSG 動作で ALET 1 は基本空間から、最後に制御を有した部分空間に制御を移す。

【 0 0 8 5 】 ステップ 83 は、基本空間を表わす ALET = 0 を検査する。もし指定 ALET = 0 であれば、宛先 ASTE (DASE) は基本空間である。DUCT の SA は 0 (部分空間は活動状態ではない) にセットされ、CR1 は BASTEO で ASTE からの STD をロードされる。そして PSW 及び他の CR ロードのためにステップ 99 に進む。

【 0 0 8 6 】 ステップ 86 は ALET = 1 を検査する。この値は実行中のグループの現在又は最後の部分空間を表わす。もし指定 ALET = 1 であれば、ステップ 91 で DUCT の S SASTEO は BSG の DASE にセットされる。

23

【 0 0 8 7 】 ステップ 92 は有効な ASTE であることを検査し、ステップ 93 は SSASTEO によりアドレス指定された DASTEO 内の ASTESN が DUCT 内の SSASTESN に等しいことを検査する。ステップ 92 で ASTE が無効であるか又はステップ 93 で不一致がある場合は、ステップ 94 で例外が通知される。

【 0 0 8 8 】 前記無効又は不一致がなければ、ステップ 97 で、 SA は部分空間が活動状態であることを示す 1 にセットされ、 CR1 は宛先 ASTE の STD の適切なビットをロードされる。(ESA/390 Principle of Operation に記述されている) 空間切替え事象及び記憶変更事象を制御する CR1 のビット 0 及び 24 は、それらが部分空間グループ内の基本空間及び全ての部分空間に当てはまると見なされるので不变のままである。制御は PSW 及び他の CR をロードするためにステップ 99 に進む。もし ALET の値が 0 でもなく 1 でもなければ、ステップ 87 に進み、 BSG 命令で指定された AR2 内の ALET を用いて宛先 ASTE が ART により発見される。ステップ 88 で、もし DUCT 内のアドレス DASTEO=BAST EO であれば、その基本空間は分岐の目標であり、前述のように制御はステップ 98 に進む。

【 0 0 8 9 】 ステップ 95 は宛先空間がグループの一部でありますか別の基本空間ではないことを検査する。グループの一部ではない場合又は異なる基本空間である場合は、ステップ 94 で例外が通知される。ステップ 96 は、 SSASTEO フィールドを BSG の DASTEO に、そして SSASTESN フィールドを DASTE の ASTESN の値にセットすることにより、部分空間実行の DUCT フィールドを設定する。そして、前述のように、制御はステップ 97 に進む。

【 0 0 9 0 】 ステップ 99 で、 R1 は復帰アドレス指定モード及びアドレスを与える PSW ビット 32-63 をロードされ、 PSW ビット 32-63 は新しいアドレス指定モード及び命令アドレスとして BSG で指定された R2 からロードされ、第 2 の空間が新たに第 1 の空間にセットされ、第 2 の ASN が PASN にセットされる。 CR7 を CR1 に等しくし、 SA SN を PASN に等しくすることは、宛先プログラムがその呼び出し元を直接アドレス指定する能力を取り除き、アドレス空間のあいだのアドレス分離を可能にする。

【 0 0 9 1 】 BSG 命令の動作が終了するとプロセッサは PSW 及び CR1 により指定された次の命令の実行に移る。

【 0 0 9 2 】 [ 図 9 ]

部分空間グループ機能は次の ESA/390 命令: PC(プログラム呼出し)、 PT(プログラム移行)、 PR(プログラム復帰)、 SSAR(第 2 の ASN 設定)及び LASP (アドレス空間パラメータ・ロード) の動作に影響する。

【 0 0 9 3 】 新たな定義は概略次のように行われる。第 1 のアドレス空間又は第 2 のアドレス空間としてアドレス空間が設定されるごとに、もしアドレス空間が(そのアドレス空間の STD 内の G ビット = 1 で示すように)部分空間グループの一部であり、そして部分空間が( DUCT 内の SA=1 で示すように)活動状態であれば、アドレス

10

20

30

40

50

24

空間の ASTEO は DUCT 内の BASTEO と比較される。もし SSASTE のビット 0 が 0 であれば、 DUCT 内の SSASTESN と SSASTE 内の ASTESN とが比較される。もし両者が等しければ、( CR1 又は CR7 内の) アドレス空間の STD のビット 1-23 及び 25-31 は SSASTE 内の STD の同じビットにより置き換えられる。

【 0 0 9 4 】 PC 、 PT 、 PR 及び SSAR については、もし SSASTE 内のビット 0 が 1 であれば、又はもし DUCT 内の SSASTESN と SSASTE 内の ASTESN とが等しくなければ、例外が認められる。 LASP については、例外は認められない。代わりに、設定されるアドレス空間に応じて状態コードがセットされる。第 1 の空間が設定中のとき状態コード 1 がセットされ、第 2 のアドレス空間が設定中のとき状態コード 2 がセットされる。

【 図面の簡単な説明】

【 図 1 】 2 レベルの仮想アドレス変換テーブルで得られ、本発明により使用できるグループ関連の共通(及び非共通)の実アドレス指定を表わす図である。

【 図 2 】 1 レベルの仮想アドレス変換テーブルで得られ、本発明により使用できるグループ関連の共通(及び非共通)の実アドレス指定を表わす図である。

【 図 3 】 MVS/ESA のようなオペレーティングシステムで設定しうるような部分空間グループのアドレス指定能力を示す図である。

【 図 4 】 図 3 の部分空間グループの基本空間及び 1 つの部分空間を、図 1 の方法で、共通ページ・テーブルの使用により設定できる方法を示す図である。

【 図 5 】 図 3 の部分空間グループの基本空間及び 1 つの部分空間を、図 2 の方法で、共通ページ・テーブル・エンティリの使用により設定できる方法を示す図である。

【 図 6 】 基本空間及び関連部分空間を有するグループを示す仮想アドレス指定構成、並びに基本空間と部分空間の間のプログラム制御のアドレス空間切替えの際に用いる関連制御ブロックの例を示す図である。

【 図 7 】 空間がプロセッサ制御レジスタによりアドレス指定された部分空間グループの一部であるかどうかを示すフィールドを有する TID 及び指名可能装置制御テーブルの例として ESA/390 STD を示す図である。

【 図 8 】 BSG (部分空間グループ内の分岐) 命令の CPU (中央処理装置) 実行の流れ図である。

【 図 9 】 部分空間環境で正しく動作するために要求される、プログラム呼出し、プログラム転送、プログラム復帰、二次 ASN セット及びアドレス空間パラメータ・ロード命令での特別な検査を示す図である。

【 符号の説明】

20	基本空間セグメント・テーブル (B-ST)
21	部分空間セグメント・テーブル 1 (S-ST1)
22	部分空間セグメント・テーブル 2 (S-ST2)
30	基本空間ページ・テーブル (B-PT)
31	部分空間ページ・テーブル (S-PT1)
32	部分空間ページ・テーブル (S-PT2)

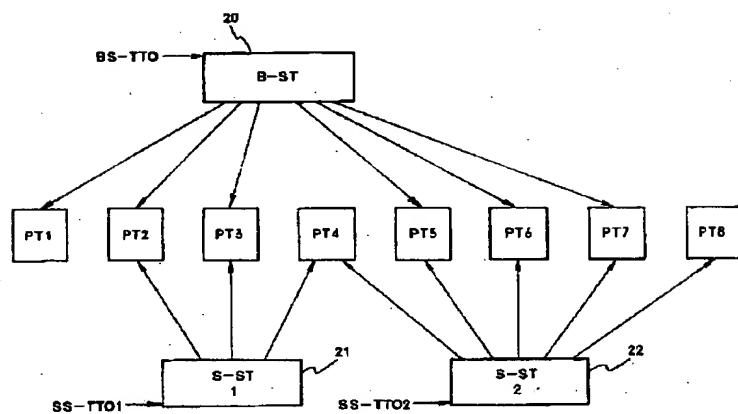
25

40	列／基本空間／BS
40A	セグメント・テーブル
40B	セグメント・テーブル
40C	ページ・テーブル
41	列／部分空間／SS1
41A	セグメント・テーブル
41B	セグメント・テーブル
41C	ページ・テーブル
42	列／部分空間／SS2
43	列／アドレス空間／SS3

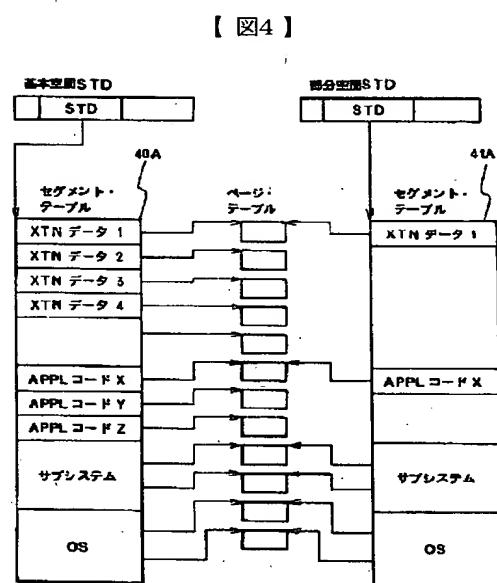
10

50	タスク制御ブロック (TCB)
52	DUCT PIR(指名可能装置制御テーブル・ポイン
タ)	
61	R1
62	R2
66	DU-AL(指名可能装置アクセス・リスト)
71	ASTE
72	フィールド
73	ワード

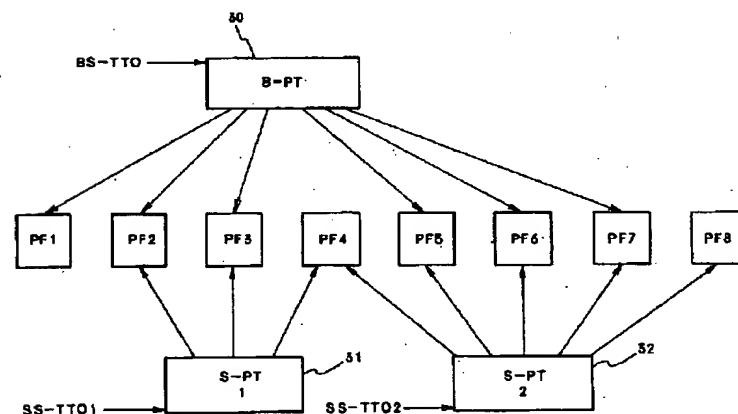
【 図1 】



【 図2 】



【 図4 】



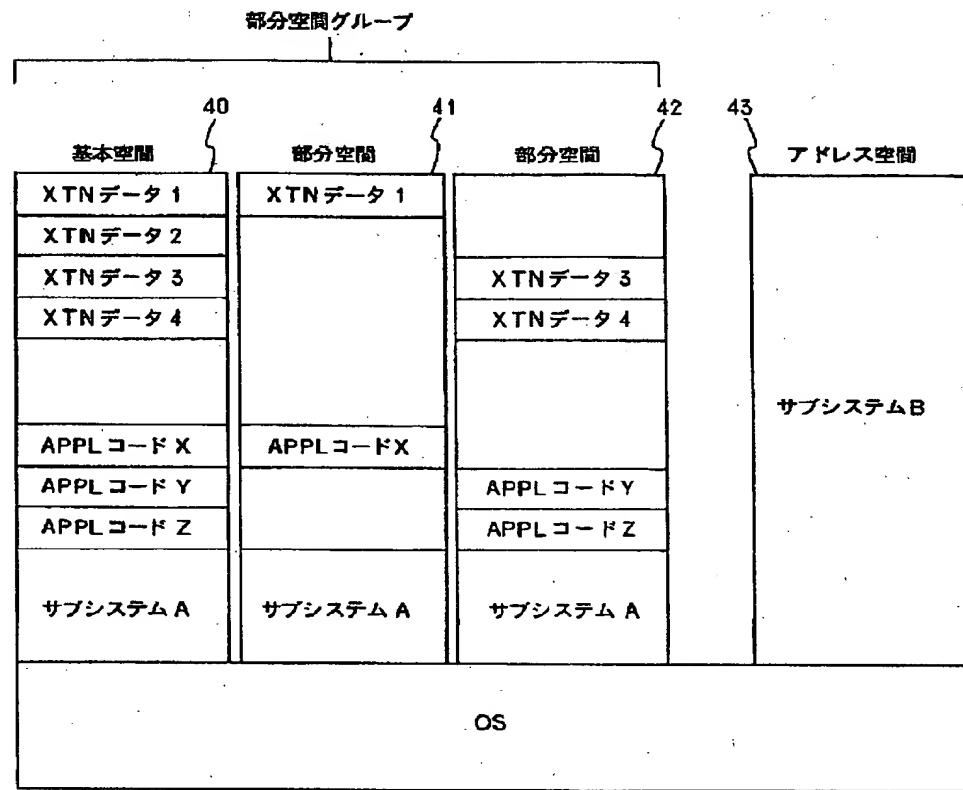
【 図7 】

セグメント・テーブル指定 (STD)

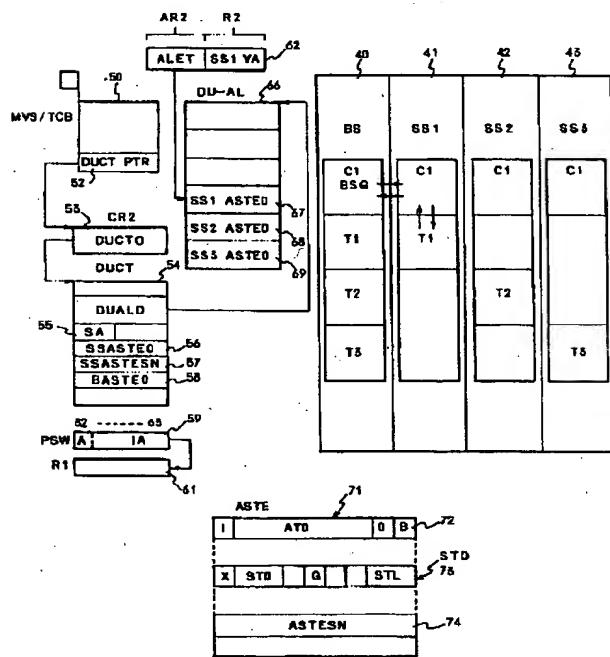
	STO		G	P	S		STL

$G = 1 \Rightarrow$  STDにより指定されたアドレス空間は  
部分空間グループ内にある

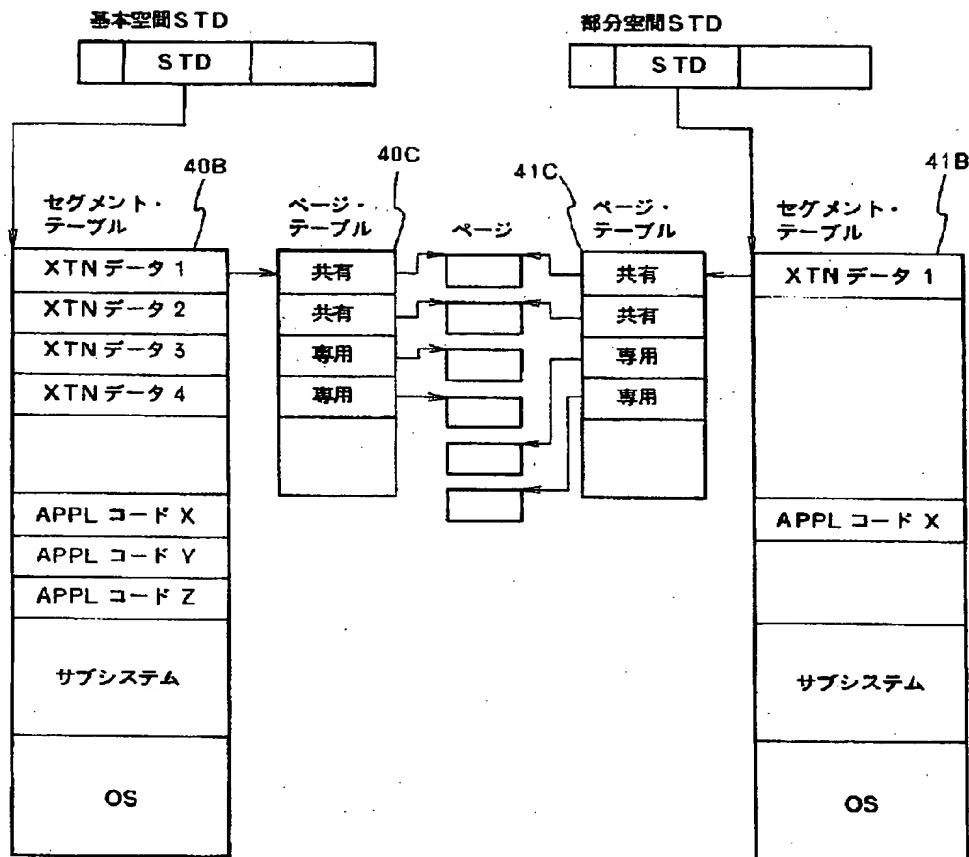
[ 図3 ]



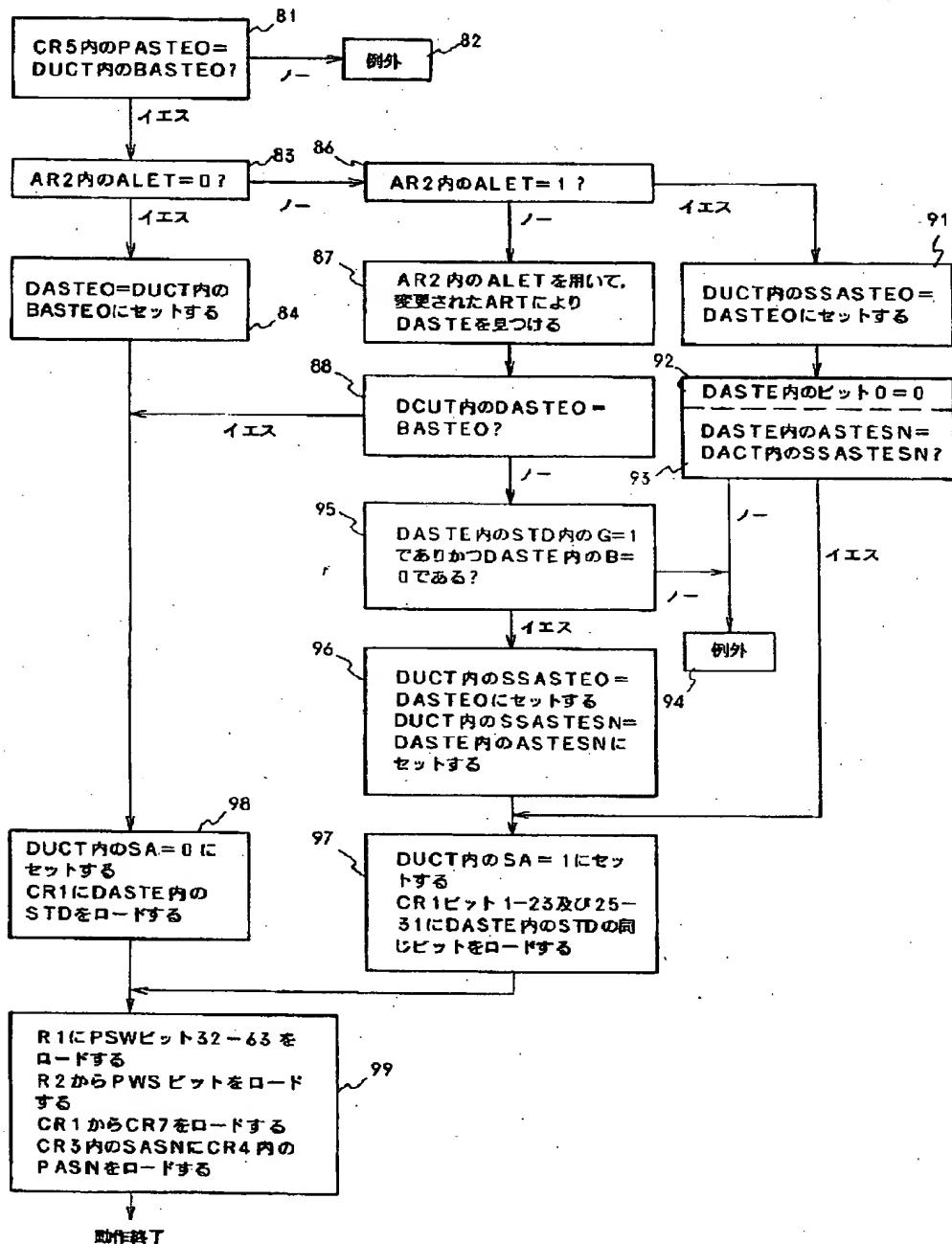
【図6】



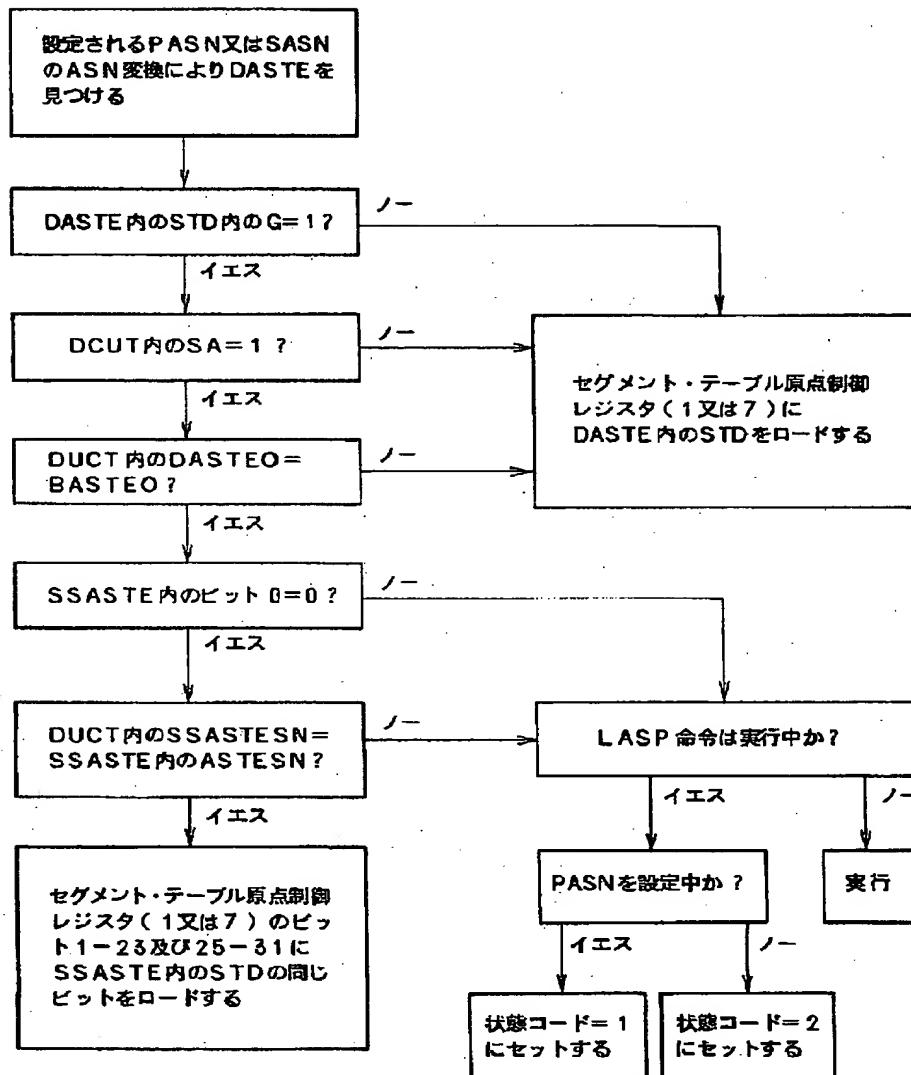
【図5】



【 図8 】



【 図9 】



フロントページの続き

(72)発明者 ジェフレイ・アラン・フレイ  
アメリカ合衆国12524、ニューヨーク州フ  
ィッシキル、グリーンヒル・ドライブ 24  
エー  
(72)発明者 ケネス・アーネスト・ラムベック  
アメリカ合衆国12603、ニューヨーク州ボ  
ーキプシー、デイジー・レーン 7

(72)発明者 キャスパー・アンソニー・スカルズイ  
アメリカ合衆国12601、ニューヨーク州ボ  
ーキプシー、アパートメント 7イー、ア  
カデミー・ストリート 160  
(72)発明者 バースカー・シンハ  
アメリカ合衆国12590、ニューヨーク州ワ  
ツピングジャーズ・フォールズ、ケンダル・  
ドライブ 19